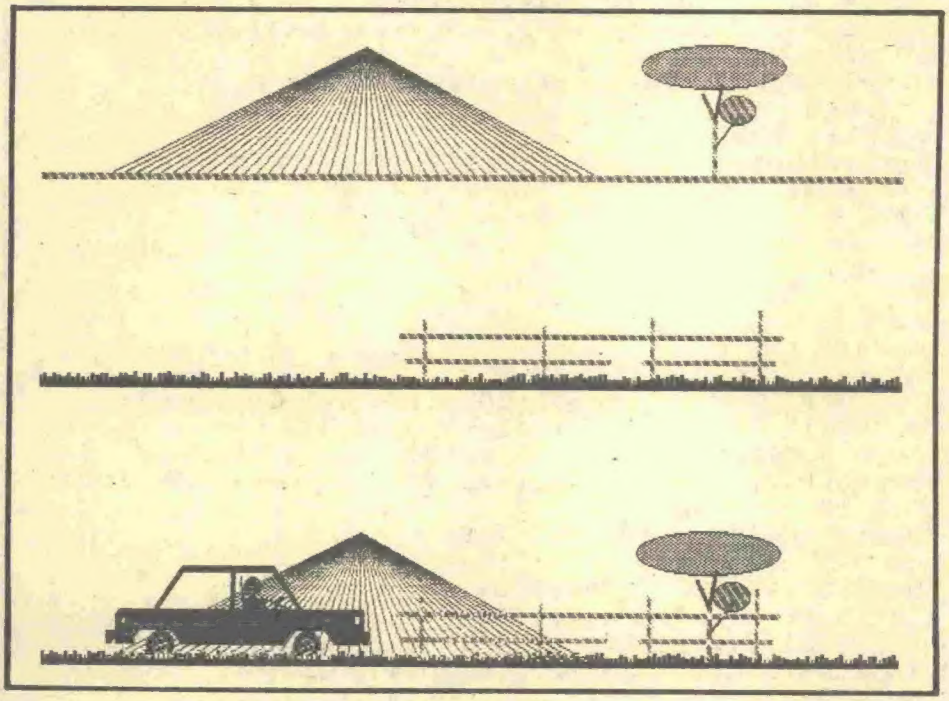
การประยุกต์ใช้ เทอร์โบปาสคาล 4.0-5.0 กับ CAI

ตอน 4



อาจหาญ สัตยารักษ์

หวังว่าผู้อ่านคงสนุกกับกราฟิกบนปาสคาล 4.0 ไม่มากก็น้อย คิดว่าจากตอนที่ 1,2 และ 3 นั้นคงจะช่วยให้ท่านมีความเข้าใจในการใช้คำสั่งกราฟิกมากขึ้น ตอนนี้เราจะมาศึกษาคำสั่งอื่น ๆ ต่อ

โดยตอนที่ 4 นี้ผู้เขียนจะพูดถึงคำสั่งที่ช่วยจัดการกับจอภาพเป็นจุดใหญ่ เพราะคำสั่งเหล่านี้ช่วยให้เราเคลื่อนย้ายภาพ, ลบบางส่วนของภาพตลอดจนช่วยทำภาพเคลื่อนไหวได้อีกด้วย แน่นอนอาจมีผู้อ่านบางท่านคิดว่าเราสามารถทำให้ภาพเคลื่อนที่หรือเคลื่อนไหวได้ด้วยคำสั่ง PutImage ที่กล่าวไปแล้วในตอนที่ผ่านมา แต่ถ้าท่านลองใช้คำสั่ง PutImage นี้กับภาพที่มีขนาดใหญ่จะพบว่าการเคลื่อนที่นั้น ๆ ไม่มีความนุ่มนวลเลย เพราะกว่าที่ภาพจะถูกแทนที่หมดผู้ดูก็จะเห็นการแทนที่ภาพกวาดจากด้านบนลงมาข้างล่างอย่างชัดเจน

วันนี้เราจะปรับปรุงการสร้างภาพเคลื่อนที่ให้ดูดีขึ้น...มาติดตามต่อไป

คำสั่งแรกที่จะศึกษา คือ

```
SetViewPort (x1, y1, x2, y2:word;
-Clip : boolean);
```

คำสั่งเพื่อใช้กำหนดขอบเขตของจอภาพใหม่ สำหรับให้ประโยชน์เช่น กำหนดให้ภาพที่สร้างไปแสดง ณ ตำแหน่งใหม่ของจอภาพ หรือกำหนดขอบเขตจอภาพไว้เพื่อจะลบจอภาพได้เป็นส่วน ๆ ซึ่งจะใช้คู่กับคำสั่งที่จะกล่าวถึงต่อไป

ค่าของ x1, y1, x2, y2 คือตัวแปรตัวเลขจำนวนเต็ม (ชนิด word) ดังนั้นแม้จะกำหนดให้ค่าเป็นลบ ปาสคาลจะยอมรับเป็นค่าต่ำสุดคือ 0 เท่านั้น ค่าจึงเป็นไปตามเงื่อนไขดังนี้

1. ค่า x1, y1 คือมุมบนซ้ายของขอบเขตจอภาพที่ต้องการกำหนด ซึ่งปาสคาลจะถือว่าจุดนี้เป็นจุด 0,0 ของจอภาพที่กำหนด เช่น ถ้าเรากำหนดค่า x1,y1 เป็น 200,300 เมื่อใช้คำสั่ง Putpixel(10,10,Blue); ก็จะกลายเป็นการลงจุด 210,310 แทน

2. ค่า x2,y2 คือมุมล่างขวาของขอบเขตที่ต้องการกำหนด

ค่าของตัวแปรทั้ง 4 ต้องมีค่าเป็นดังนี้ 0 <= x1 < x2 และ 0 <= y1 < y2

3. Clip เป็นตัวแปรชนิดบูลีน กำหนดได้สองลักษณะคือ True และ False หรือ Clipon และ Clipoff โดยมีการทำงานดังนี้

True หรือ Clipon กำหนดแล้วการวาดภาพ หรือลบภาพ จะทำเฉพาะในส่วนที่ถูกกำหนดขอบเขตไว้เท่านั้น

False หรือ Clipoff จะสามารถวาดภาพ หรือลบภาพเลยจากขอบเขตที่กำหนดไว้นั้นได้

การใช้งานจริงจึงเป็นดังตัวอย่างต่อไปนี้

```
SetViewPort(40,30,719,347,True);
SetViewPort(200,120,600,300,False);
SetViewPort(20,0,400,200,Clipon);
SetViewPort(300,200,600,300,
Clipoff);
```

ใช้คำสั่ง SetViewPort(0,0, GetmaxX,GetMaxY,Clipon); เมื่อต้องการให้กลับมาใช้ขอบเขตของจอภาพตามปกติ

คำสั่งที่ 2 คือ ClearViewPort ซึ่งเป็นคำสั่งที่ใช้คู่กับคำสั่งที่ 1 เมื่อต้องการลบบางส่วนของจอภาพออกไป เช่นต้องการลบจอภาพในขอบเขต (200,200,500,300) ก็ใช้คำสั่งดังนี้

```
SetViewPort(200,200,500,300,Clipon);
ClearViewPort;
```

ทำให้สามารถลบจอภาพส่วนดังกล่าวออกไปได้ทันที ซึ่งสามารถสร้างเป็นชุดคำสั่งได้เช่น

รูปที่ 1
โปรแกรมการใช้คำสั่ง SetView Port, ClearPort

```
{$I INITGR.PAS}

PROCEDURE Ram(count,x,y,Cx,Cy,Sp:integer; mode:byte);
var xx,yy:integer;
begin
 xx:=0;
 for yy:=1 to count do
   begin
     setfillstyle(9,1);  Bar3D(x+xx,y,x+xx+Cx,y+Cy,3,topon);
     setfillstyle(1,1);  Bar(x+xx+2,y+2,x+xx+Cx-2,y+Cy-2);
     if mode>0 then begin
      setfillstyle(6,1); pieslice(x+xx+9,y+7,0,360,4);
                 end;
     if mode=2 then begin
      setfillstyle(6,1); pieslice(x+xx+9,y+cy-7,0,360,4);
                 end;
      xx:=xx+Sp;
   end;
end; { Ram }

PROCEDURE Card;
 var k,x:integer;
 begin
  setfillstyle(10,1);          Bar3D(30,0,300,100,2,topon);
  Bar(200,100,293,110);        setfillstyle(6,1);
  RectAngle(30,0,300,100);     RecTangle(200,100,293,110);
  x:=203;  Moveto(293,110); lineto(295,107); lineto(295,100);
    for k:=0 to 10 do  begin
            bar(x,100,x+4,110); x:=x+8;
                       end;
  Ram(7,34,25,11,25,19,0);    Ram(9,34,65,9,25,15,0);
  Ram(1,80,4,9,18,15,0);    Ram(7,170,47,9,20,15,0);
  Ram(6,200,75,9,20,15,0);   Ram(1,34,14,4,6,1,0);
  Ram(2,170,74,4,23,5,0);    Ram(2,170,23,4,20,5,0);
  Ram(2,190,5,24,33,30,1); Ram(1,110,10,44,10,5,0);
  Ram(1,250,10,44,4,5,0);    Ram(1,275,30,13,36,5,0);
end; { Card }

{ ============= SETVIEWPORT & CLEARVIEWPORT =============}
begin
opengraph;
 bar(0,0,719,347);
 SetviewPort(200,100,500,300,clipon); ClearViewPort;
 delay(1000);
 SetviewPort(0,0,719,347,clipon);     ClearViewPort;
 SetViewPort(290,10,600,347,Clipon); Card;
 SetViewPort(360,55,620,305,Clipoff); Card;
 SetViewPort(70,119,719,347,Clipon); Card;
 SetViewPort(20,175,400,207,Clipoff); Card;
 SetViewPort(380,175,719,347,Clipon); Card;
 SetViewPort(0,0,GetMaxX,GetMaxY,Clipon); Card;
 Waittogo;
 closegraph;
end.
```

```
Procedure Erase(x1,y1,x2,y2:word);
begin
    SetViewPort(x1,y1,x2,y2,Clipon);
    ClearViewPort;
    SetViewPort(0,0,GetmaxX,GetMaxY,
    Clipon);
end;
```

ซึ่งจะนำไปใช้ลบส่วนของจอภาพได้ง่าย ๆ ขึ้นได้อีกทางหนึ่ง

โปรแกรมในรูปที่ 1 จะเป็นตัวอย่างในการศึกษาคำสั่งทั้งสองนี้ โดยเป็นการสร้างภาพของการ์ด (สำหรับเครื่องคอมพิวเตอร์) ซึ่งตามชุดคำสั่ง Card นั้นจะเห็นว่าเป็นการกำหนดจุดที่คงที่ไว้ แต่เมื่อกำหนดขอบเขตของจอภาพใหม่ก็สามารถจะกำหนดให้ภาพไปเกิดที่ตำแหน่งอื่น ๆ ของจอภาพได้ ดูผลของการวาดภาพทั้งหมดได้จากรูปที่ 2

คำสั่งที่ 3 และคำสั่งที่ 4 ที่จะกล่าวถึงต่อไปนี้ คือ

SetVisualPage และ SetActivePage

สำหรับกำหนดเกี่ยวกับจอภาพอีกแบบหนึ่ง แต่เป็นกำหนดว่าจะใช้จอภาพใดแสดงภาพ ใช้จอภาพใดวาดภาพ กล่าวคือ ปาสคาล 4.0 นั้นมีจอภาพกราฟิกในการทำงานซ้อนกันอยู่ 2 จอ ภาพ คือ กราฟิกหน้า 0 และกราฟิกหน้า 1 ดังนั้นจึงใช้เลข 0 และ 1 เป็นตัวกำหนดว่าจะใช้กราฟิกหน้าใดในการวาดภาพหรือแสดงภาพ

เราสามารถใช้กราฟิกหน้า 0 แสดงภาพให้ผู้ชมดู แล้วแอบไปเขียนภาพที่กราฟิกหน้า 1 ไว้ จากนั้นก็พลิกกราฟิกหน้า 1 มาให้ดูใหม่พร้อมกับแอบไปสร้างภาพที่กราฟิกหน้า 0 ต่อ ซึ่งจะทำอย่างนี้ไปได้เรื่อย ๆ คล้าย ๆ กับการแปรอักษรในการแข่งขันกรีฑานั่นเอง

โดยมีรายละเอียดดังนี้

SetVisualPage สำหรับกำหนดการใช้กราฟิกหน้าใดแสดงภาพ เช่น SetVisualPage(0); ก็เป็นการเลือกให้ภาพที่แสดงเป็นภาพของกราฟิกหน้า 0 และ SetVisualPage(1); ก็เป็นการใช้กราฟิกหน้า 1 มาแสดงภาพให้ดูแทน

SetActivePage ใช้สำหรับกำหนดว่ากราฟิกหน้าใดจะเป็นส่วนวาดภาพ ซึ่งถ้ากำหนดไว้เป็นคนละค่ากับ SetVisualPage แล้ว การวาดภาพนั้นจะยังมองไม่เห็น เช่นใช้ SetActivePage (0); ก็หมายถึงกำหนดกราฟิกวาดภาพเป็นกราฟิกหน้า 0 เช่นเดียวกับ SetActivePage(1); เป็นการใช้กราฟิกหน้า 1 มาวาดภาพแทน

โดยปกติเมื่อไม่มีการกำหนดปาสคาลจะใช้กราฟิกหน้า 0

การทำให้ภาพเคลื่อนที่

ด้วยคำสั่ง SetVisualPage และ SetActivePage นี้เราสามารถสร้างภาพเคลื่อนไหวได้ มาดูปัญหากันสัก 2 เรื่อง ดังนี้

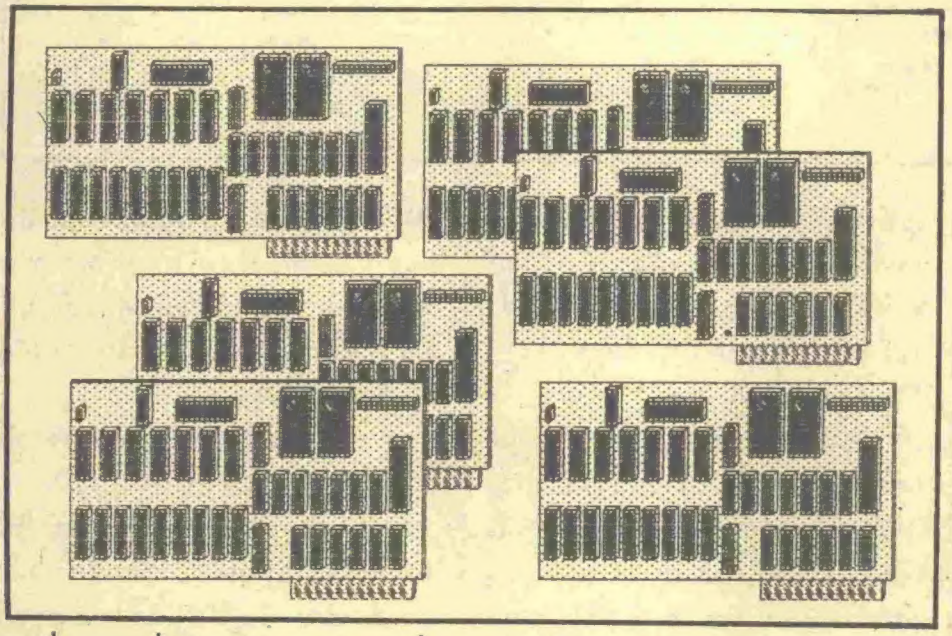
ปัญหาที่หนึ่ง เมื่อเราขว้างก้อนหินก้อนหนึ่งลงในน้ำก็จะเกิดคลื่นน้ำกระจายออกจากจุดขว้างเป็นหน้าคลื่นวงกลมโตขึ้นเรื่อย ๆ

ปัญหาที่สอง เมื่อเราต้องการให้แสดงภาพรถยนต์คันหนึ่งแล่นไปบนถนน ซึ่งอาจเป็นถนนที่ว่างเปล่า หรืออาจเป็นถนนที่มีทั้งฉากหลังและฉากหน้ามาประกอบด้วยก็ได้ (ซึ่งแน่นอนการแสดงจะต้องมีความยากขึ้นด้วย)

เราสามารถใช้คำสั่งให้แสดงภาพเหล่านี้ได้หลายวิธี ลองมาดูเล่นที่ปัญหาแรกก่อนโดยจะเขียนได้เป็น 2 แบบ ดังโปรแกรมในรูปที่ 3 โดยแบบแรกในชุดคำสั่ง Wave I เป็นการวาดและลบตามปกติ ส่วนแบบที่สองนั้นใช้คำสั่ง SetVisualPage และ SetActivePage เข้ามาช่วยด้วย คือชุดคำสั่ง Wave II แต่ไม่ได้พิมพ์ภาพมาให้ดู เพราะต้องการให้ศึกษาการทำงานบนจอภาพมากกว่า

โปรแกรมนี้จะเรียกไฟล์ initgr.pas (กล่าวถึงในตอนที่ 1) เข้ามาใช้งานด้วย

ที่นี้มาดูปัญหาที่สองบ้าง โดยจะมาดูในประเด็นที่มีฉากหน้าประกอบเพียงอย่างเดียวก่อน ซึ่งผู้เขียนจะใช้ชุดคำสั่ง Pic (จากตอนที่ 3) ซึ่งเป็นภาพรถยนต์มาใช้งานร่วมกับไฟล์ initgr.pas เช่นเดิม พร้อมกับเขียนโปรแกรมขึ้นมาใหม่ (โปรแกรมในรูปที่ 4) อัน



รูปที่ 2 ภาพที่ได้จากโปรแกรมในรูปที่ 1

รูปที่ 3
โปรแกรมการเกิดคลื่นคลวงกลม

```
{$I INITGR.PAS}

PROCEDURE WaveI;
var x,y,c:integer;
begin
   x:=0; c:=1;
   for y:=0 to 40 do
    begin
     setcolor(1); circle(370,170,x);
     setcolor(0); circle(370,170,x);
     x:=x+3;
    end;
    setcolor(1); circle(370,170,x);
    waittogo;
end;  {WaveI }

PROCEDURE WaveII;
var x,y,c:integer;
begin
    SetFillStyle(1,0); Bar(0,0,719,347);
    x:=0; c:=1;          TextShow(2,7,'h');
    OuttextXY(180,50,'SetVisualPage and SetActivePage');
    SetActivePage(1);
    OuttextXY(180,50,'SetVisualPage and SetActivePage');
   for y:=0 to 20 do
    begin
     SetVisualPage(0);    SetActivePage(1);
     setcolor(1); circle(370,170,x+3);
     setcolor(0); circle(370,170,x-3);
     SetVisualPage(1);     SetActivePage(0);
     setcolor(1); circle(370,170,x+6);
     setcolor(0); circle(370,170,x);
     x:=x+6;
    end;
    waittogo;
end; {WaveII }

{====================  WAVE ==================}
begin
  OpenGraph;
   WaveI;
   WaveII;
  CloseGraph;
end.
```

ประกอบชุดคำสั่ง 4 ชุด คือรั้วไม้ (ชุดคำสั่ง Fence), ภาพคนขับรถ พร้อมกับปรับปรุงภาพของล้อรถให้ดูดีขึ้นกว่าเดิม (ชุดคำสั่ง Driver) และชุดคำสั่งให้เกิดภาพเคลื่อนที่ 2 วิธี คือแบบไม่ใช้คำสั่ง SetVisualPage และคำสั่ง SetActivePage แบบหนึ่ง กับใช้คำสั่งทั้งสองนี้อีกแบบหนึ่ง (ชุดคำสั่ง Car_I, Car_II) ซึ่งมีการจำภาพและลงภาพด้วย PutImage เหมือนกัน

อย่าลืมว่าโปรแกรมในรูปที่ 4 จะทำงานได้ต้องมีคำสั่ง {$I initgr.pas} และเรียกชุดคำสั่ง Pic เข้ามาใช้งานด้วย ขอให้ศึกษาการทำงานบนจอภาพเองว่าแบบใดในสองแบบนี้ที่ให้ผลของการทำงานที่สมบูรณ์กว่า

คงมองออกแล้วว่าวิธีที่ 2 นั้นแสดงภาพเคลื่อนไหวที่เป็นธรรมชาติมากกว่าวิธีที่ 1 การแสดงภาพด้วยการ PutImage(x-5,66,PicCar^,1); เป็นการกำหนดค่าแบบ xor ดังนั้นเมื่อกำหนดครั้งแรก ณ จุดใดแล้วแสดงทับ ณ จุดเดิมอีกครั้งก็จะเป็นการลบภาพที่ลงนั้นออกไป โดยพื้นจะคงเดิม จากโปรแกรมใช้วิธีวาดภาพรั้วทับรถยนต์ที่เคลื่อนที่ทุก ๆ ครั้ง จึงมองเห็นเป็นว่ารั้วอยู่ด้านใกล้กับคนดู

หากเรามีฉากหลังด้วยเราจะทำอย่างไร

การที่จะทำภาพเคลื่อนไหวที่มีฉากหลังและฉากหน้าด้วยนั้น ต้องคำนึงถึงหลักความจริงที่ว่า ฉากหลังจะถูกบังด้วยวัตถุที่เคลื่อนที่ผ่าน และฉากหน้า

รูปที่ 4 โปรแกรมสร้างภาพรถยนต์เคลื่อนที่ 2 แบบ

```
PROCEDURE Fence;
begin
  SetFillStyle(6,1);     bar(503,108,614,110);
  bar(300,94,620,96);    bar(305,108,476,110);
  bar(320,86,322,124);   bar(420,89,422,124);
  bar(510,84,512,124);   bar(600,80,602,124);
  SetFillStyle(1,1);
end; {  Fence }

PROCEDURE Driver;
begin
  Pieslice(175,80,0,360,7);
  Bar(167,87,185,92);    Bar(173,85,177,87);
  line(166,87,165,75);   line(167,87,166,75);
  line(191,93,198,86);   line(192,93,199,86);
  line(193,93,200,86);   line(194,93,201,86);
  line(196,90,215,95);   line(197,90,216,95);
  line(185,87,196,90);   line(186,88,197,90);
  Pieslice(98,113,0,360,11);
  Pieslice(221,113,0,360,11);
  setcolor(0);           circle(98,113,6);
  circle(221,113,6);     setcolor(1);
end; { Driver }

PROCEDURE Car_I;
Var  PicCar:pointer;
     Size:word;
     k,x :integer;
Begin
  Bar(0,121,719,124);  Fence;
  pic(53,120,3);       Driver;
  Size:=ImageSize(50,66,280,120);
  GetMem(PicCar,Size);
  GetImage(50,66,280,120,PicCar^);
  PutImage(50,66,PicCar^,1);
   x:=50;
   for k:=1 to 80 do
    begin
     PutImage(x,66,PicCar^,1);
     PutImage(x,66,PicCar^,1);  Fence;
     X:=x+6;
    end;
 waittogo;
end;  {Car_I }

PROCEDURE Car_II;
Var  PicCar:pointer;
     Size:word;
     k,x :integer;
begin
  SetVisualPage(0);   SetActivePage(1);
  SetFillStyle(1,0); bar(0,0,719,347);
  SetFillStyle(1,1);
  Bar(0,121,719,124);  Fence; pic(53,120,3);
  Driver;
  Size:=ImageSize(45,66,275,120);
  GetMem(PicCar,Size);
  GetImage(45,66,275,120,PicCar^);
  SetVisualPage(1);   SetActivePage(0);
   PutImage(50,66,PicCar^,0); Fence;
  x:=50;
   for k:=1 to 49 do
    begin
     SetVisualPage(0);   SetActivePage(1);
       PutImage(x-5,66,PicCar^,1);
       PutImage(x+5,66,PicCar^,1);   Fence;
     SetVisualPage(1);    SetActivePage(0);
       PutImage(x,66,PicCar^,1);
       PutImage(x+10,66,PicCar^,1);  Fence;
     x:=x+10;
    end;
 waittogo;
end; { Car_II }

{=============== Main ==================}
begin
 opengraph;
  Car_I;
  Car_II;
 closegraph;
end.
```

รูปที่ 6 โปรแกรมสร้างภาพรถยนต์เคลื่อนที่แบบที่มีฉากหลังประกอบด้วย

```
Var  PicCar:array[1..4] of pointer;
     PicBack,PicFore:pointer;
     Size:word;

PROCEDURE ForeGround;
var k,x,y:integer;
begin
  SetFillStyle(1,0); bar(0,0,719,347); SetFillStyle(1,1);
  SetFillStyle(6,1);     bar(503,108,614,110);
  bar(300,94,620,96);    bar(305,108,476,110);
  bar(320,86,322,124);   bar(420,89,422,124);
  bar(510,84,512,124);   bar(600,80,602,124);
  SetFillStyle(1,1);     Bar(0,121,719,124);
   x:=0;  SetLineStyle(3,$FF,3);
   for k:=0 to 180 do   begin
         y:=round(int(random(5))); line(x,120,x,120-y);
         x:=x+4;
                        end;
         SetLineStyle(0,$FF,1);
  Size:=ImageSize(0,66,719,120);  GetMem(PicFore,Size);
  GetImage(0,66,719,120,PicFore^);
end; { ForeGround }

PROCEDURE BackGround;
var k,x:integer;
begin
  SetFillStyle(1,0);    bar(0,0,719,347);
  SetFillStyle(6,1);    Bar(560,90,563,124);
  line(560,90,566,70);  line(561,90,567,70);
  line(562,90,568,70);  line(560,90,550,75);
  line(562,90,551,75);  line(563,90,552,75);
  ellipse(560,60,0,360,60,12);
  ellipse(580,84,0,360,15,8);
  SetFillStyle(9,1);     floodfill(560,58,blue);
  SetFillStyle(6,1);     floodfill(580,80,blue);
  line(563,103,580,88);  line(563,104,580,89);
  Bar(0,121,719,124);    SetFillStyle(1,1);
  x:=0;
    for k:=0 to 30 do  begin
      Moveto(60+x,120); lineto(270,50);  lineto(460-x,120);
      x:=x+10;             end;
    Size:=ImageSize(0,30,719,120);  GetMem(PicBack,Size);
  GetImage(0,30,719,120,PicBack^);
end; { BackGround }

PROCEDURE GetPicCar;
var k,x:integer;
begin
   Bar(0,121,719,124);    pic(53,120,3);  Driver;
   x:=359; setcolor(1);
   for k:=1 to 4 do begin
        Pieslice(98,113,0,360,11);    Pieslice(221,113,0,360,11);
        setcolor(0);
        circle(98,113,6);                  circle(221,113,6);
        Pieslice(98,113,x,x+2,6);          Pieslice(98,113,x-90,x-92,6);
        Pieslice(98,113,x-180,x-182,6);    Pieslice(98,113,x-270,x-272,6);
        Pieslice(221,113,x,x+2,6);         Pieslice(221,113,x-90,x-92,6);
        Pieslice(221,113,x-180,x-182,6);   Pieslice(221,113,x-270,x-272,6);
        Size:=ImageSize(50,66,280,120);    GetMem(PicCar[K],Size);
        GetImage(50,66,280,120,PicCar[K]^);
        setcolor(1);    X:=x-22;
                    end;
end;   { GetPicCar }
```

รูปที่ 5 (ต่อ)

```
PROCEDURE Car_III;
var     k,x,c :integer;
begin
  SetVisualPage(0);  SetActivePage(1);
  SetFillStyle(1,0); bar(0,0,719,347); SetFillStyle(1,1);
  Bar(0,121,719,124);
  BackGround;                ForeGround;
  PutImage(0,30,PicBack^,0);  PutImage(0,66,PicFore^,1);
  SetVisualPage(1);     SetActivePage(0);
  GetPicCar;
  PutImage(0,30,PicBack^,0);
  PutImage(0,66,PicCar[1]^,2);    PutImage(0,66,PicFore^,1);
  x:=0;  c:=1;
   for k:=1 to 49 do
    begin
       SetVisualPage(0);    SetActivePage(1);
       PutImage(0,30,PicBack^,0);
       PutImage(x,66,PicCar[c]^,2);    PutImage(0,66,PicFore^,1);
     SetVisualPage(1);     SetActivePage(0);
       PutImage(0,30,PicBack^,0);
       PutImage(x+5,66,PicCar[c+1]^,2);    PutImage(0,66,PicFore^,1);
     x:=x+10; c:=c+2; if c=5 then c:=1;
    end;
    Sound(300); delay(1000); nosound;
 waittogo;
end; { Car_III }

{=============== Main ==================}
begin
 opengraph;
  Car_III;
 closegraph;
end.
```

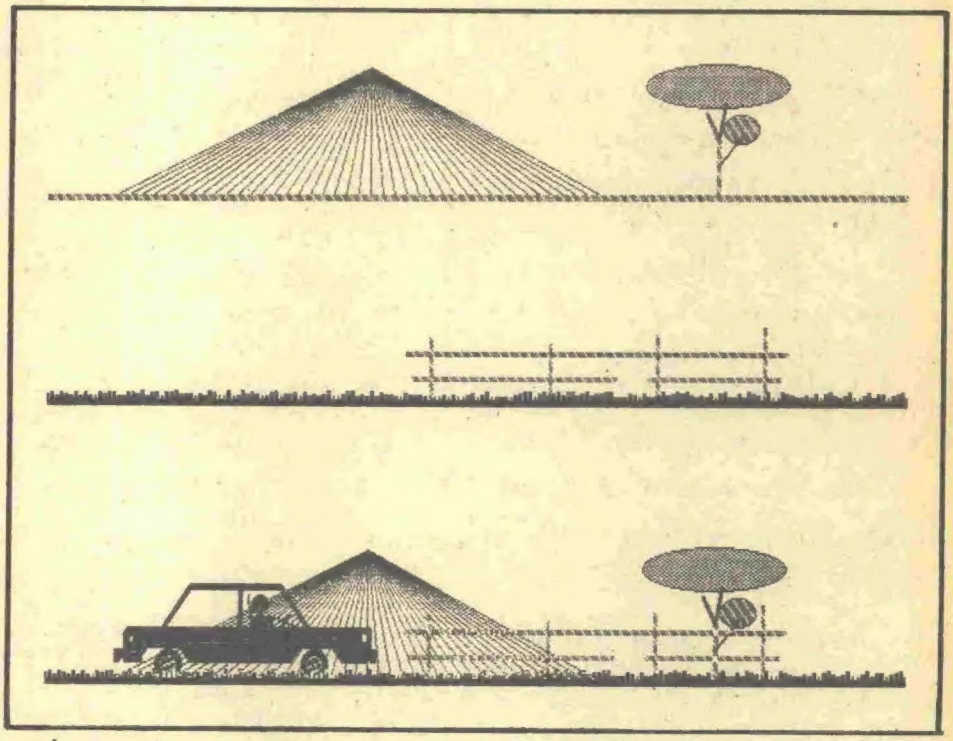
จะต้องดูออกว่าอยู่ข้างหน้าจริง ๆ จึงจะทำให้ดูการเคลื่อนที่นั้น ๆ ตรงตามธรรมชาติ

สมมติว่าเรามีฉากหลังเป็นภาพภูเขาและต้นไม้ มีฉากหน้าเป็นต้นหญ้าและรั้ว เราจะทำภาพเคลื่อนที่ของรถยนต์ให้ดูสมจริงได้ดังนี้

1. จัดให้กราฟิกหน้า 0 เป็นส่วนแสดงภาพด้วยคำสั่ง SetVisualPage(0); และให้กราฟิกหน้า 1 เป็นส่วนวาดภาพด้วยคำสั่ง SetActivePage(1);

2. วาดรูปฉากหลังก่อน อาจจะวาดภาพตรง ๆ ไปเลย หรือใช้คำสั่ง PutImage แบบ mov (รหัส 0) ซึ่งเป็นการแทนที่ภาพลงไปโดยทำให้ภาพเดิมหายไปก็ได้

3. วาดรูปรถยนต์ หรือใช้คำสั่ง PutImage แบบ or (รหัส 2) ซึ่งเป็น



รูปที่ 6 ภาพฉากหลัง ฉากหน้า และ ภาพการเคลื่อนที่จากโปรแกรมในรูปที่ 5

การแทนที่แบบตรวจสอบจุดว่าลงจุดหรือไม่ลงจุดโดยคงสภาพการลงจุดไว้เหมือนเดิม (ถ้าเป็นจอสีจะเปลี่ยนสี)

4. วาดภาพฉากหน้าทับลงไปหรือใช้คำสั่ง PutImage แบบ xor (รหัส 1) ก็ได้

5. กลับมาใช้คำสั่ง SetVisualPage(1); เพื่อให้แสดงภาพ และแอบไปเขียนกราฟิกอีกหน้าด้วยคำสั่ง SetActivePage(0); โดยวาดภาพฉากหลังและฉากหน้าแบบเดิม แต่เลื่อนภาพของรถไปข้างหน้าอีกนิดหนึ่ง (ทำตามข้อ 2,3 และ 4)

6. กลับไปทำข้อ 1-5 จนกว่าจะได้ตำแหน่งที่พอใจ

โปรแกรมในรูปที่ 5 คือโปรแกรมที่จะทำได้ตามแนวคิดที่กล่าวมาทั้งหมด โดยใช้งานร่วมกับไฟล์ initgr.pas และชุดคำสั่ง Pic, ชุดคำสั่ง Driver พร้อมกับเขียนชุดคำสั่งใหม่อีก คือ

ForeGround สำหรับวาดภาพฉากหน้า

Back Ground สำหรับวาดภาพฉากหลัง

GetPicCar สำหรับเก็บภาพรถยนต์ 4 ภาพที่มีลักษณะของล้อที่ต่างกัน เพื่อจะไปใช้ต่อเนื่องกันทำให้ดูเสมือนว่าล้อรถหมุนไปพร้อมกับการเคลื่อนที่

Car_III ส่วนที่จะทำการวาดภาพแสดงการเคลื่อนที่

รูปที่ 6 คือภาพฉากหลัง ฉากหน้า และภาพการเคลื่อนที่ในช่วงแรก ๆ ส่วนตอนที่รถกำลังเคลื่อนที่ผ่านรั้วนั้น ภาพจะเป็นอย่างไร ๆ ต้องศึกษาเองที่จอภาพจะได้ความเข้าใจที่ดีกว่า

คงได้คำสั่งใช้งานในตอนนี้แค่ 4 คำสั่งเท่านั้น แต่เป็นคำสั่งที่มีประโยชน์มาก ๆ ทีเดียว

เนื่องจากส่วนอื่น ๆ ยาวมากอยู่แล้ว ฉบับนี้จึงขอเสนอโปรแกรมฟิสิกส์สั้น ๆ โดยของดคำอธิบายและการพิมพ์ภาพมาให้ดูด้วย เพราะคิดว่าผู้อ่านคงจะศึกษาได้ไม่ยาก เป็นเรื่องอะไรก็ขอให้ศึกษาได้จากโปรแกรมในรูปที่ 7 และส่วนที่ผู้เขียนทำไว้ในโปรแกรมชุดนี้ยังไม่สมบูรณ์ เพราะเพียงกำหนดแนวทางการเขียนโปรแกรมไว้เท่านั้น จึงอยากให้ผู้อ่านลองนำความรู้เท่าที่ได้ศึกษามาเขียนโปรแกรมในรูปที่ 7 ให้มีความสมบูรณ์ยิ่งขึ้น เช่นอาจแสดงการรวมกันของคลื่น 2 คลื่น, แสดงการสะท้อนของคลื่น ฯลฯ

พบกันใหม่...ฉบับหน้า

รูปที่ 7 โปรแกรมแสดงการเคลื่อนที่ของคลื่น

```
{$I initgr.pas }

Var WavePic1,WavePic2:pointer;
    Size:word;

PROCEDURE Wave;
var   y:word;
      k:integer;
      x:real;
begin
  Moveto(0,100);  x:=0;
  for k:=0 to 1440 do
   begin
     y:=round(20*Sin(k/180*pi));
     lineto(round(x),100-y);
     x:=x+0.5;
   end;
  size:=ImageSize(0,80,710,140);
  GetMem(WavePic2,Size);
  GetImage(0,80,710,140,Wavepic2^);
  size:=ImageSize(711,80,719,140);
  GetMem(WavePic1,Size);
  GetImage(711,80,719,140,Wavepic1^);
end; { Wave }

PROCEDURE Move;
begin
 Repeat
  SetVisualPage(0);  SetActivePage(1);
  PutImage(0,80,WavePic1^,0);
  PutImage(8,80,WavePic2^,0);
  GetImage(0,80,710,140,Wavepic2^);
  GetImage(711,80,719,140,Wavepic1^);
  SetVisualPage(1);  SetActivePage(0);
  PutImage(0,80,WavePic1^,0);
  PutImage(8,80,WavePic2^,0);
  GetImage(0,80,710,140,Wavepic2^);
  GetImage(711,80,719,140,Wavepic1^);
 until keypressed;
end; { Move }

begin
   OpenGraph;
     Wave;
     Move;
   WaitToGo;
   Closegraph;
end
```